REMARKS/ARGUMENTS

In the Office Action of July 29, 2004, Claims 1-3, 5, 8-14, and 16-25 are rejected under 35 U.S.C. 102(e) as being anticipated by U.S. Patent 6,202,150 issued to Young et al. ("Young et al."); Claim 4 is rejected under 35 U.S.C. 103(a) as being unpatentable over Young et al. in view of Richard Stevens (TCP/IP, illustrated, Vol. 1) ("Stevens"); and Claims 6-7 and 15 are rejected under 35 U.S.C. 103(a) as being unpatentable over Young et al. in view of U.S. Patent No. 5,799,086 issued to Sudia ("Sudia").

1. Rejection of Claims 1-3, 5, 8-14, and 16-25 under 35 U.S.C. 102(e)

Before discussing the rejected claims, it is worthwhile to describe and contrast the very different applications addressed by Applicants' invention and that of Young et al. in order to more clearly understand why Young et al. fails to anticipate Applicants' rejected claims as amended.

With regard to Applicants' invention, software vendors distribute their software as binary executable code, which is machine readable, but not human readable. To enhance or correct bugs in the software, however, programmers work with the source code of the software, which is human readable, but not machine readable. Attachment A defines the term "source code" and illustrates its relationship with the binary "executable code" of a software program.

Since customers only receive binary executable code, they must rely on vendors to make any enhancements or correct bugs in the software program. When vendors go bankrupt or are otherwise unable to perform these responsibilities, the customers may find themselves in a predicament, because of their heavy reliance on the proper operation of the licensed software in their businesses.

If customers had copies of the source code, however, they could enhance or correct bugs themselves. Vendors do not want to provide their source code to customers, however, for security reasons as well as the fact that they generate ongoing revenue by selling maintenance services commonly including such enhancements and bug fixes to their customers.

To balance these competing interests, source code escrows have been established so that a trusted third party can hold the source code in escrow, and only release it to customers in the event that the vendor is unable to meet its obligations regarding enhancements and bug fixes. Because of the fragile nature of the media upon which the source code is stored, special environmentally controlled facilities are required to be maintained by source code escrow companies to guarantee media integrity. Security is also a serious matter, since source code theft could severely damage a vendor's business. This further adds to their expense.

Applicants' invention is therefore directed to a novel, low cost approach to performing source code escrows. Rather than providing the source code on media to the escrow

holder, only a decryption key is provided. Thus, expensive, environmentally controlled facilities are not required. Also, theft of the decryption key, by itself, is not useful, because the customer holds the encrypted source code, not the escrow holder.

In contrast, Young et al. addresses the problem of law enforcement agencies needing to "wire-tap" cryptographic communications between criminals. See, e.g., Col. 1, lines 43-47. This is the same application addressed in other cited prior art in the Office Action, such as Sudia (see, e.g., Col. 6, lines 17-23 of Sudia), and Y. Desmedt, "Securing Traceability of Ciphertexts – Towards a Secure Software Key Escrow System," Eurocrypt '95, pp. 147-157, Springer-Vertag, 1998 (see, e.g., abstract).

In this application, recipients are not prevented in any manner from accessing messages or attachments sent to them. The messages are sent encrypted with the recipient's public key, and the recipient decrypts the encrypted messages using the recipient's private key. See, e.g., Col. 1, lines 16-39.

Since there are legitimate privacy concerns with merely giving law enforcement and other government agencies access to private keys, private key escrows have been proposed. See, e.g., Col. 1, lines 49-51. The private keys may then be shared with law enforcement agencies in the event of a court authorized wire tap. See, e.g., Col. 1, lines 51-53. Note that there is generally no need to share the private keys with the recipient, since it is the recipient's private key that is being exchanged. The only time that the

recipient would need the private key from the private key escrow is if the recipient somehow lost the recipient's private key. See, e.g., Col. 1, lines 53-54.

Accordingly, referring to amended Claim 1 now, it is clear from the above discussion that Young et al. does not teach or suggest the element of "providing said software key to an escrow holder who is under instructions to provide said software key to said recipient upon satisfaction of a release condition, wherein said software key is otherwise unavailable to the recipient at any time." As explained above, in Young et al., the software key is the recipient's private key, therefore there is no reason to provide the software key to the recipient unless the recipient has for some reason lost its private key. In any event, this would not be a case where the software key was "otherwise unavailable to the recipient at any time."

Further, Young et al. does not disclose encrypting source code and providing the encrypted source code to a recipient. In this regard, note that the definition of source code as provided, for example, in Attachment A, does not encompass the signal information which includes the user's public key and certificate of recoverability as described in Young et al. Although the encrypting and providing of the encrypted source code to a recipient may be performed through cryptographic communications, the "mere possibility" of such activity without a description of such or an inherent need to do so is believed to be improper grounds for a rejection under 35 U.S.C. 102(e) and the result of impermissible hindsight reasoning.

Accordingly, Claim 1 is believed to be patentable under 35 U.S.C. 102(e) over Younger et al. for the foregoing reasons.

Claims 2-3, 5 and 8-9 are also believed to be patentable under 35 U.S.C. 102(e) over Younger et al. since they depend from Claim 1, and as such, are believed to be patentable for at least the same reasons as stated in reference to Claim 1.

Further, with respect to Claim 3, Young et al. does not disclose generating binary executable code and providing the encrypted source code and the binary executable code to a recipient. Although such activity may be performed through cryptographic communications, the "mere possibility" of such activity without a description of such or an inherent need to do so is believed to be improper grounds for a rejection under 35 U.S.C. 102(e) and the result of impermissible hindsight reasoning.

Also, with respect to Claim 8, using a computer readable medium to provide the encrypted source code and binary executable to a recipient would totally defeat the purpose of Young et al. to provide a means for law enforcement agencies to "wire-tap" cryptographic communications between criminals, and therefore, in addition to such activity not being taught in Young et al., there could also be no suggestion of such activity.

Amended Claim 10 is also believed to be patentable under 35 U.S.C. 102(e) over Younger et al. for essentially the same reasons as stated in reference to Claim 1.

Claims 11 and 12 are also believed to be patentable under 35 U.S.C. 102(e) over Younger et al. since they depend from Claim 10, and as such, are believed to be patentable for at least the same reasons as stated in reference to Claim 10, as well as, in the case of Claim 11, the reasons stated in reference to Claim 3.

Amended Claim 13 is also believed to be patentable under 35 U.S.C. 102(e) over Younger et al. for essentially the same reasons as stated in reference to Claim 1.

Claims 14 and 15 are also believed to be patentable under 35 U.S.C. 102(e) over Younger et al. since they depend from Claim 13, and as such, are believed to be patentable for at least the same reasons as stated in reference to Claim 13, as well as, in the case of Claim 14, the reasons stated in reference to Claim 8.

Amended Claim 16 is also believed to be patentable under 35 U.S.C. 102(e) over Younger et al. for essentially the same reasons as stated in reference to Claim 1.

Claim 17 has been cancelled and its limitations incorporated into Claim 16.

Claim 18 is also believed to be patentable under 35 U.S.C. 102(e) over Younger et al. since it depends from Claim 16, and as such, is believed to be patentable for at least the same reasons as stated in reference to Claim 16, as well as, the reasons stated in reference to Claim 3.

Claims 19 and 20 have been cancelled and their limitations incorporated into Claim 16.

Amended Claim 21 is also believed to be patentable under 35 U.S.C. 102(e) over Younger et al. for essentially the same reasons as stated in reference to Claim 1.

Claim 22 has been cancelled and its limitations incorporated into Claim 21.

Claim 23 is also believed to be patentable under 35 U.S.C. 102(e) over Younger et al. since it depends from Claim 21, and as such, is believed to be patentable for at least the same reasons as stated in reference to Claim 21, as well as, the reasons stated in reference to Claim 3.

Claims 24 and 25 have been cancelled and their limitations incorporated into Claim 21.

2. Rejection of Claim 4 under 35 U.S.C. 103(a)

Stevens is only used in the Office Action to show the common known use of transferring files over the Internet using the file transfer protocol (FTP). There is no contention in the Office Action that Stevens teaches or suggests any of the limitations of Claim 1, and it is believed that there is no such teaching or suggestion in Stevens regarding the elements of Claim 1 that are neither taught nor suggested by Young et al.

Accordingly, Claim 4 is believed to be patentable under 35 U.S.C. 103(a) since it depends from Claim 1, and as such, is believed to be patentable for at least the same reasons as stated in reference to Claim 1, as well as those stated in reference to Claim 3, and further since neither Young et al. nor Stevens, alone or in combination, teach or suggest all of the limitations of Claim 4, including its base claim and intervening claim.

3. Rejection of Claims 6-7 and 15 under 35 U.S.C. 103(a)

Sudia is only used in the Office Action to show the common known use of sending information via email. There is no contention in the Office Action that Sudia teaches or suggests any of the limitations of Claim 1 or Claim 13, and it is believed that there is no such teaching or suggestion in Sudia regarding the elements of Claim 1 or Claim 13 that are neither taught nor suggested by Young et al.

Accordingly, Claims 6-7 are believed to be patentable under 35 U.S.C. 103(a) since they depend from Claim 1, and as such, are believed to be patentable for at least the same reasons as stated in reference to Claim 1, and further since neither Young et al. nor Sudia, alone or in combination, teach or suggest all of the limitations of Claims 6-7, including their base claim and intervening claims.

Claim 15 is also believed to be patentable under 35 U.S.C. 103(a) since it depends from Claim 13, and as such, is believed to be patentable for at least the same reasons as stated in reference to Claim 13, and further since neither Young et al. nor Sudia, alone or in

combination, teach or suggest all of the limitations of Claim 15, including its base claim and intervening claim.

Claims 1-16, 18, 21, and 23 are pending in the application. Claims 17, 19-20, 22, and 24-25 have been cancelled. Reconsideration of the rejected pending claims is respectfully requested, and an early notice of their allowance earnestly solicited.

Respectfully submitted,

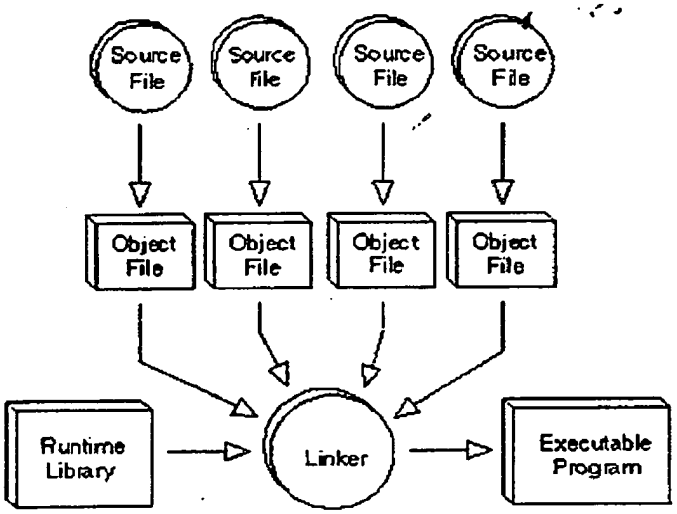
Dated: September 29, 2004

Victor H. Okumoto
Registration No. 35,973
Office Phone: (510) 792-1112

Attachment A

(Source: http://www.webopedia.com/TERM/s/source_code.html; September 28, 2004)

source code



Program instructions in their original form. The word *source* differentiates code from various other forms that it can have (for example, object code and *executable code*).

Initially, a programmer writes a program in a particular programming language. This form of the program is called the *source program*, or more generically, *source code*. To execute the program, however, the programmer must translate it into machine language, the language that the computer understands. The first step of this translation process is usually performed by a utility called a compiler. The compiler translates the source code into a form called object code. Sometimes the object code is the same as machine code; sometimes it needs to be translated into machine language by a utility called an assembler.

Source code is the only format that is readable by humans. When you purchase programs, you usually receive them in their machine-language format. This means that you can execute them directly, but you cannot read or modify them. Some software manufacturers provide source code, but this is useful only if you are an experienced programmer.